Pervasive Pheromone-Based Interaction with RFID Tags

MARCO MAMEI & FRANCO ZAMBONELLI
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia

Despite the growing interest in pheromone-based interaction to enforce adaptive and context-aware coordination, the number of deployed systems exploiting digital pheromones to actually coordinate the activities of situated autonomous agents is still very limited. In this paper, we present a simple, low-cost and general-purpose implementation of a pheromone-based interaction mechanism for pervasive environments. This is realized by making use of RFID tags to store digital pheromones, and by having humans or robots spread/sense pheromones by properly writing/reading RFID tags populating the surrounding physical environment. We exemplify and evaluate the effectiveness of our approach via an application for object-tracking. This application allows robots and humans to find "forgotten-somewhere" objects by following pheromones trails associated with them. In addition, we sketch further potential applications of our approach in pervasive computing scenarios, discuss related work in the area, and identify future research directions.

Categories and Subject Descriptors: I.2.11 [Artificial Intelligence]: Multiagent Systems; C.2.4 [Computer-Communication Systems]: Distributed Systems; C.3 [Special Purpose and Application-based Systems]: Smartcards.
General Terms: Algorithms, Design, Experimentation
Additional Key Words and Phrases: Stigmergy, Pervasive Computing, RFID Tags.

---

1. INTRODUCTION

Pheromone-based interaction, adopted by social insects to coordinate their activities [BonDT99], has recently inspired a vast number of researches in pervasive and distributed computing systems [BabMM02, MenT03, SveK04, ParBS05]. In these works, autonomous and mobile application agents (whether mobile software agents, humans carrying on a PDA, or autonomous robots) interact with the surrounding world and with each other by leaving and sensing artificial pheromones trails, digital analogues of chemical markers, in the environment. Pheromones, by encoding application-specific information in a distributed way and by uncoupling the activities of application agents, enable to enforce adaptive and context-aware coordination activities [Par97].

Despite the growing interest in pheromone-based interaction, the number of implemented systems exploiting pheromones for coordinating the activities of distributed agents situated in pervasive computing scenarios is still very limited. The great majority of the proposals have only been simulated [BonDT99, MenT03], only few of them have been concretely implemented by deploying pheromones in shared virtual data spaces [ParBS05], other few realize pheromones by means of ad-hoc physical markers such as special ink or metal dust [SveK04] (see related work section).

These approaches – especially those based on network-accessible data spaces [ParBS05] – could be profitably complemented by a completely distributed pervasive infrastructure that could store pheromone values without requiring connectivity to remote data spaces. In this context, we propose a novel approach exploiting RFID technology [Wan04, Wan06] to enforce pheromone-based interaction in pervasive computing scenarios. The key idea of our approach (earlier presented at a conference [MamZ05]), is to exploit RFID tags dispersed in an environment as a sort of distributed memory in which to store digital pheromones. RFID reader devices, carried by humans or by robots, could deploy pheromone trails in the environment simply by writing pheromone values in the RFID tags around. Also, they could sense such pheromone-trails by simply reading pheromone values in RFID tags nearby. Clearly, such an approach is extremely low cost ad not intrusive, as RFID tags will soon or later be present in any case in any environment.

This kind of infrastructure could backup network based approaches in presence of network unavailability (as in the case of disaster scenarios) to create truly autonomic systems capable of surviving connectivity problems. In this sense, we think that the unique improvement of our proposal over current approaches is that it can work without any kind of network infrastructure and with extremely cheap hardware that will be widespread in the future.

Relying on our simple yet flexible approach, a wide range of application scenarios based on pheromone interaction can be realized, ranging from monitoring and supporting of everyday human activities [Phi04], multi-robot coordination [SveK04], and impromptu coordination in challenged scenarios [MamQZ06]. Here, after having illustrated our approach, we detail and evaluate an application to easily find – by following proper RFID pheromone trails – everyday objects forgotten somewhere in our homes. Results extracted from both tests on a real implementation with RFID-reader-equipped PDAs and robots, and from simulation experiments show that our approach is very effective in enforcing context-awareness and in facilitating coordination in physical environments, though it also exhibits some limitations induced by the limited resource capabilities of RFID tags.

The remainder of this paper is organized as follows. Section 2 briefly introduces pheromone-based interaction. Section 3 describes the RFID technology and presents some basic services required to implement our approach. Section 4 details our approach to implement pervasive pheromone-based interaction. Section 5 illustrates the object tracking application example. Section 6 exploits such application to evaluate the

effectiveness of our approach and to identify its limitations. Section 7 sketches several additional application scenarios which can take advantage of our approach. Section 8 discusses related work. Section 8 concludes and outlines open research directions.

2. PHEROMONE-BASED INTERACTION

Ants and other social insects interact by spreading chemical markers (i.e., pheromones) as they move in the environment, and by being directed in their actions by the perceived concentrations of pheromones. This simple mechanism of local interactions mediated by the environment, called stigmergy, enables ants to globally self-organize their collective activities in a seemingly intelligent way and to adaptively act in an unknown environment. Since such adaptive context-aware behavior is exhibited despite the very limited abilities of individuals in acquiring and cognitively processing contextual information, systems of social insects are said to be characterized by "swarm intelligence", to emphasize the difference with "individual" intelligence [BonDT99, Par97].

The classical example to show the power of pheromone-based interaction is ant foraging. Ants in a colony, when in search for food, leave the nest and start wandering around. When some food is found, they start spreading a pheromone and try to get back to the nest, thus creating a trail leading to the food source. When an ant is looking for some food, it can indirectly exploit the past experience of other ants by following an existing pheromone trail to reach previously discovered food sources. This also contributes to reinforce the pheromone trail in the ant will spread pheromones in its turn. To some extent, the environment becomes a sort of distributed repository of contextual information holding the information about all the paths to the discovered food sources. The natural tendency of the pheromones to evaporate if not reinforced, allows the pheromone network to remain up-to-date and to adapt to changing conditions: when some ants discover a shorter path to food, longer paths tend to be abandoned and disappears; analogously, when a food source is extinguished, the corresponding pheromone trail disappear because no longer reinforced [BonDT99].

Despite its simplicity, pheromone-based interaction presents several features that make it suitable in a lot of distributed and pervasive applications:

1. it completely decouples agent (i.e., ant) interactions, which occur indirectly via the mediation of pheromones. This is a very desirable feature in open and dynamic scenarios where agents do not know each other in advance and can come and go at any time;

2. it naturally supports application-specific context awareness, in that pheromones provide agents with an expressive application-specific representation of their operational environment (e.g. pheromones provide a representation of the environment in terms of paths leading to food sources);
3. it naturally supports adaptation of activities, in that pheromones represent a contextual information that, when no longer updated, tend to vanish;
4. the algorithms underlying pheromone-based interaction are simple and involve only local interactions (each ant locally deposits and follows pheromones without any clue – and associated burden/complexity – of being involved in a distributed task).

Given these features it is not surprising that several research proposals, in area as diverse as routing in networks [BonDT99], P2P computing [BabMM02, MenT03], robotics [ParBS05, SveK04], self-assembly [SheS02], and (as in our approach) pervasive computing, incorporate and exploit pheromone-based interaction mechanisms.

3. RFID INFRASTRUCTURE

The technology of Radio Frequency Identification (RFID) is at the core of our approach for deploying digital pheromones in an environment. RFID tags are tiny wireless radio transceivers that can be attached to objects as small as a watch or a toothbrush. Tags can be purchased off the shelf, cost roughly €0.20 each and, being battery-free, they do not have power-exhaustion problems. Each tag is marked with a unique identifier and provided with a tiny memory (up to some Kb) allowing to store data in the form of array of bytes.

Suitable devices, called RFID readers, can be interfaced with portable computers and can be used to access RFID tags by radio for read or write operations (i.e., to read/write specific bytes in the RFID memory). The tags respond or store data using the power scavenged from the signal coming from the RFID reader. RFID readers divide into short- and long-range depending on the distance within which they can access RFID tags, from a few centimeters (short-range readers) up to some meters (long-range readers).

So far, RFID technology has been mostly exploited as an alternative (more robust and flexible) to optical barcodes for automated identification of goods, as it may be required in anti-thefts systems and in logistics [Bor04]. More recently, their potential of applicability in pervasive computing is getting recognized, and a variety of applications and infrastructures exploiting RFID technology are being proposed [Phi04, FloL05,

HsiF05]. However, to the best of our knowledge, our proposal is the first one that suggests exploiting RFID tags to bring pheromones in the real world.

3.1 Scenario Assumptions

Our approach requires a scenario in which the operational environment is densely enriched with RFID tags, and in which human users and robots carry/embed some handheld computing devices provided with a RFID reader. In the next future *(i)* many household objects and furniture will be RFID-tagged before purchase and *(ii)* handheld devices provided with embedded RFID read and write capabilities will have an increasing diffusion (for instance, the Nokia 5140 phone can be already equipped with a RFID reader). These factors will make our assumption become a *de facto* situation, and will make our approach become directly deployable at nearly zero cost.

It is fair to report that the current EPC standard (www.epcglobalinc.org), adopted by most major retailers, uses read-only RFID tags. Moreover, privacy concerns may force tags attached to objects to be deactivated out of the retailer. We think that, both these two constraints, that would make our infrastructure unfeasible, are temporary and will disappear in the future. On the one hand, there are a number of researches actively investigating novel writable tags with large storage capacity (www.technologyreview.com/printer_friendly_article.aspx?id=17182). On the other hand, research on security and privacy issues in RFID is becoming more and more active, and some solutions addressing these concerns without destroying the tags are emerging [RieCT06], lasecwww.epfl.ch/~gavoine/rfid. Ultimately, applications – like the one presented in this paper – that are made possible by the availability of writable tags will justify the increased costs and/or privacy concerns.

A number of services – there included pheromone deployment, described in the next section – can be realized by exploiting RFID tags attached to fixed locations (e.g. doors, corridors, etc.) and to unlikely-to-be-moved objects (e.g., beds, washing machines, etc.). We generically refer to these tags as *location-tags*, to distinguish them from other tags that are likely to be move often (e.g., those attached to clothes and small objects).

Other than the unique ID, *location-tags* can contain simple information about the location (e.g., "what=office" "who=Marco"). In our implementation, we reserved 4 bytes of the RFID memory to include two simple numeric key-value pairs (associating byte values to concepts according to a simple taxonomy [MamQZ06]) to describe some facts about objects and locations. Other than the local information within a tag itself, if some WiFi connection to the Internet is available, it is possible to exploit a database to gather

additional information about the RFID existing in the environment and about what they represent. In the database, entries map the tag ID to a name, a spatial location, and any other additional information one may need to store. Thus, a user with a RFID reader and a WiFi connection, after having read a tag, can lookup its ID into the database and retrieve additional information about the object/location to which the tag is attached.

3.2 Location and Motion Estimation Using RFID Tags

A first service that can be realized on the basis of a number of *location-tags* is localization [Hah04, Sat05]. A user (or robot) provided with a RFID reader moves in an environment. A software agent controlling the reader, unobtrusively from its user, continuously detects in-range *location-tags* to infer the current location. Specifically, the agent can simply localize the user near the last read RFID tag. Such a localization can take two forms. Either the agent is provided with a map of the environment reporting the location of all the *location-tags*. This would allow the agent to actually identify its location in the environment. Or, the agent can simply take actions on the basis of the sensed spot. In the next section, for example, we will see how the agent can follow a pheromone without knowing the global map of the environment.

In addition to that simple localization, it is worth noticing that a recent work [KleP06] adopts RFID to enable robotic simultaneous localization and mapping (SLAM). This is a very interesting approach in that it further supports the use of RFID-based infrastructures in scenarios where other than localization, a map of the environment has to be built dynamically (this kind of approach well suits disaster scenarios and, in fact, it has been originally applied to the Robocup Rescue scenario).

Other than localization, RFID tags can be used to detect user motion. Basically a difference in the sensed *location-tags* indicates to the agent that the user moved. More formally (see Figure 1), let $L(t)$ be the set of *location-tags* being sensed at time t. It is easy to see that an agent can infer that the user is moving when $L(t) \neq L(t-1)$. As we will see in the next section, this is fundamental to trigger pheromone propagation.

It is worth noticing that these kind of services require a RFID reader with a rather small reading range. If, for example, the reader would be able to read tags in a 100m radius, localization would be extremely coarse. This is one of the reason, other than cost and battery-consumption, that made us prefer passive rather than active RFID tags (active tags are battery-powered devices that can be read from a large distance).

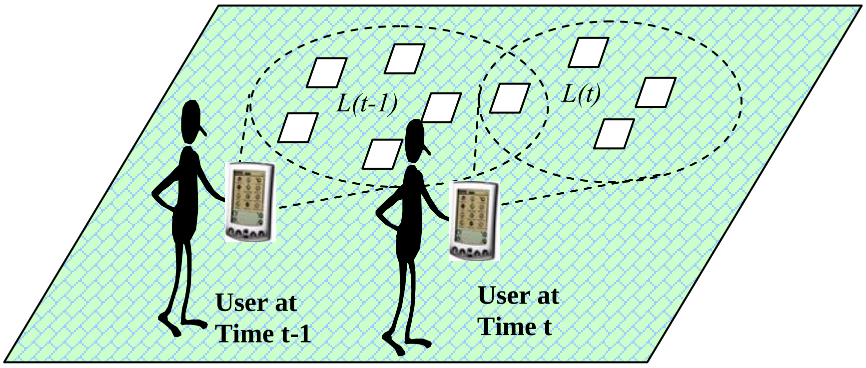


Figure 1. When the user moves, its agent gets in range with a different set of location-tags (here represented as white rectangles), and recognizes the motion.

4. DEPLOYING PHEROMONES WITH RFID TAGS

As anticipated in the introduction, pheromones are created by means of data-structures stored in RFID tags. In other words, RFID tags in the environment act as a sort of distributed environmental memory that can be used to store pheromones and to build pheromone trails.

4.1 Pheromone Deployment

The deployment of pheromones across the RFID distributed in an environment takes place via a software agent running on the portable computing devices and in control of the associated RFID reader. Whenever instructed to start spreading a pheromone *O* (in the form of a data structure consisting in a pheromone ID and in additional information detailed in the following), the agent will write *O* in the in-range *location-tags*.

In order to spread *O* around, as the location changes, the agent repeats the process by writing additional instances of such pheromone in the newly encountered *location-tags*. In particular, the agent will write *O* in all the *L(t)-L(t-1)* tags as it moves across the environment. This simple process creates digital pheromone trails distributed across the *location-tags* that the agent crossed while spreading the pheromone.

Clearly, a pheromone trail consisting of only the pheromone ID is not very useful. Indeed, most applications involve agents to follow each other pheromone trails to reach the location where the agents that originally laid down the pheromone were directed (or, on the contrary, to reach the location where they came from). Unfortunately, an agent crossing an-only-ID-trail would not be able to choose in which direction the agent that laid down that trail was directed. Thus, the data structure of each pheromone *O*, also include a 7-bits hop-counter *C(O)* associated with *O*.

More in detail, when instructed to spread a pheromone *O*, the agent initializes a *hop* counter to 0. Every time a movement is detected (*L(t)*!=*L(t-1)*), the agent reads the current value of *C*(*O)* in *L(t)*. If the tags belonging to *L(t)* do not have *O* or have a *C(O)* lower than *hop*, the agent stores the pheromone with value *hop*, otherwise the agent set *hop* to *C(O)*. Then it increments *hop* by 1 (see code in Fig. 2). The result is a pheromone trail with an ever increasing hop counter.

It is worth noticing that the fact of overwriting lower C(O) derives from the fact that in our applications the pheromone gradient is followed uphill. Thus, overwriting lower C(O) create shortcuts in the pheromone trail.

```
hop = 0;
if(L(t)!=L(t-1) {
new = read(C(O));
if(new == null || new < hop)
write(C(O)=hop);
else
hop = new;
hop++;
}
```

Figure 2. pheromone propagation

The resulting overall organization of the (limited) memory of tags is shown in Figure 3. In addition to the unchangeable unique identifier, the key-value pairs, and the Tag-Type bit (used to indicate if the tag is a *location-tag*), a 7-bit "Index" specifies how many pheromones are currently stored in the remaining part of the memory. For each pheromone, 3 byte slots are allocated. The first code the pheromone ID, the second code the *C* hop counter in the first 7 bits, while the last "*Diff*" bit specifies how to propagate the pheromone (as described in the next subsection). In addition, the third byte-slot represents a timestamp with the time the pheromone has been written. This will be used to support pheromone evaporation (as described in Subsection 3.4).

We emphasize here that the adopted organization for pheromones derives from contingent choices motivated by the very limited memory of today's RFID tags and aimed at minimizing their memory occupancy. However, since the evolution of the technology will soon make available tags with larger memories, it will be possible (without changing the basis of our approach) to enrich pheromones with expressive descriptions (e.g., several key-value pairs), and to add declarative rules related to their propagation patterns (e.g., to bound pheromone propagation to a specific area).

Despite the adopted RFID memory organization, it is worth noticing that some optimization for the memory occupancy could be employed. For example, it would be possible to avoid the hop counter and use the time step as a counter itself (i.e., the later

the time, the higher the counter). This would allow us to store a pheromone in 2 bytes rather than 3, and ultimately to store 33% more pheromones. Such kind of optimizations are in our future work agenda.

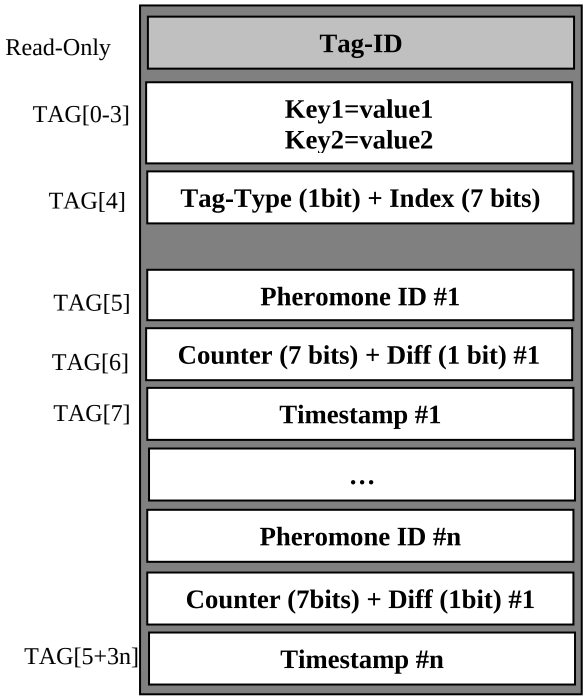


Figure 3. Memory organization of pheromones in RFID tags.

4.2 Proactive vs. Parasitic Diffusion

Unlike real-world pheromones, which diffuse around in the environment with the active support of the physical laws, the passive nature of RFID implies that a pheromone can spread only with the proactive support of some agent executing on some mobile computing device and controlling the RFID reader while it is moving.

The natural consequence is that a pheromone being spread by an agent does not actually diffuse around, but only in a specific direction. Once an agent starts spreading a pheromone, the resulting path of the pheromone trail will reproduce the path of the agent that originally diffused it. The problem of this mono-directional diffusion is that, other agents will perceive and start following a pheromone trail, only if they are lucky enough to cross it, by walking on the past steps of the agent that originally spread such pheromone. Clearly, this is not acceptable as a general solution for pheromone-based coordination.

To solve the problem, we have also integrated a form of "parasitic" diffusion of pheromones (see Figure 4). Once a pheromone trail has been diffused by an agent, other agents passing by and crossing such trail (even if having a totally different application goal) can be exploited to further support the diffusion of such a pheromone along different directions. In other words, the pheromone "parasitically" exploits the presence of a passing-by RFID reader to spread itself around along different directions (or which is the same, to branch the original mono-directional trail). When an agent diffused a pheromone parasitically, it applies an algorithm like the one in Fig. 2, but it decreases the C(O) value so as to create a path leading to the pheromone trail (when followed uphill).

The "*Diff*" bit in the pheromone data structure (see also Figure 3) has the purpose of specifying whether a pheromone should be parasitically diffused or not.

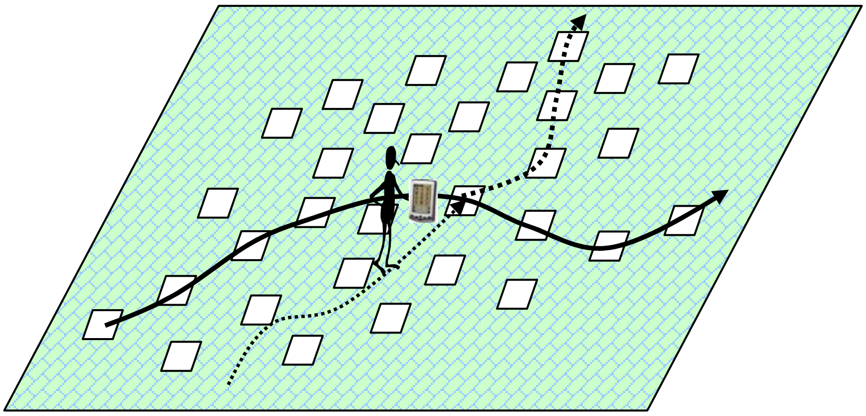


Figure 4. Parasitic Diffusion: when an agent crosses an existing pheromone trail (here represented by the chain of RFID connected via a continuous arrow), the pheromone can parasitically exploit the agent to have it branch the pheromone trail and spread the pheromone over different directions (the branched pheromone is here represented by using dashed arrows).

4.3 Pheromone Reading and Evaporation

To read pheromones, an agent trivially accesses neighbor RFID *location-tags* reading their memories in the search for pheromones with specific IDs. Clearly, this requires that the agent knows a priori what pheromone IDs correspond to its interests, i.e., via the availability of a local table of relevant pheromone IDs. We are aware such a solution is not very general and elegant. However, when more memory will be available in RFID tags to make it possible to associate a keyword-based descriptions to each pheromone, the recognition of the correct pheromone trail could occur without any need to a priori known

tables and simply by reading the descriptions associated to pheromones (in the case of available network connectivity, RFID could also store URLs pointing to further information).

Since RFID read operations are quite unreliable, the agent actually performs a reading cycle merging the results obtained at each iteration. Given the result, the agent will decide how to act on the basis of the perceived pheromone configuration and of its own application goals. For instance, in the "object-tracking" application example described in the following, we will analyze the problem of having an agent move in the environment by following uphill a pheromone trail.

Clearly, when reading a pheromone, the agent must be ensured that it represents reasonably up-to-date situations, a problem that in natural systems is solved via a process of gradual evaporation of pheromones. In our system, to realize pheromone evaporation, and since the passive nature of RFID tags does not enable them to directly enforce evaporation, we have adopted the following solution. Each pheromone is created with an associated timestamp value *T(O)* representing the time at which the pheromone *O* has been stored (see also Figure 3). To code time into the limit of a single byte, we have adopted the solution of dividing daily time into 48 ticks (1 tick = 30 minutes). This allows us not to overflow the timestamp within 5 days of use.

When pheromones are actively (instead of parasitically) deployed, the timestamp value is set to the current time – as provided by the PDA/robot clock. This naturally represents the fact that the pheromone describes an up-to-date information. On the contrary, when the pheromones are parasitically deployed, the timestamp remains the same of the original pheromone. This is because when an agent deploys the parasitic trail, it does not have more recent information, it just re-propagates "old" data.

After reading a tag, an agent checks, for each pheromone *O* it reads, whether the associated timestamp *T(O)* is, accordingly to the agent local time, older than a certain threshold *T*. If it is so, the agent deletes that pheromone from the tag. This kind of pheromone evaporation leads to two key advantages:

1. Since the data space in RFID tags is severely limited, it would be most useful to store only those pheromone trails that are important for the application at a given time; old, unused pheromones can be removed.
2. If an agent does not carry its personal digital assistant or if it has been switched off, it is possible that some actions will be undertaken without leaving the corresponding pheromone trails. This cause old-pheromone trails to be possibly out-of-date, and eventually corrupted.

In this context, it is of course fundamental to design a mechanism to reinforce relevant pheromones not to let them evaporate. With this regard, an agent spreading pheromone *O* actively, will overwrite *O*-pheromones having an older *T(O)*. From these considerations, it should be clear that the threshold *T* has to be tuned for each application, to represent the time-frame after which the pheromone is considered useless or possibly corrupted.

5. PHEROMONE-BASED OBJECT TRACKING

In this section we present a simple application we have implemented to test our approach. The application aims at facilitating the finding of everyday objects (glasses, keys, etc.) forgotten somewhere in our homes. The application allows everyday objects to leave virtual pheromone trails across our homes to be easily tracked afterwards. As simple as it can be, such an application is representative of the ways that our approach can be used, and enable us to analyze further technical issues associated with our approach.

In this application we assume that other than *location-tag* a number of objects in the environment have been tagged. We generically refer to them as *object-tags*. Such tags (as all RFID tags) include a unique ID, and a writable memory that enables to include in tags additional information about the object they refer to (e.g., “what=cup” “who=Franco”). We emphasize that the only actual – yet substantial – difference between *object-tags* and *location-tags* is that the former can be mobile and the latter can instead are assumed as fixed and can act as reference points.

Overall, the object tracking application works as follows (see Figure 5):

1. As from the assumptions, the objects to be tracked are tagged with proper *object-tags*, distinguished from the *location-tags* identifying locations in the environment.
2. Users (or robots) are provided with a handheld computing device, connected to a RFID reader, and running the object tracking application.
3. The application can detect, via the RFID reader, *object-tags* carried on by the user. Exploiting the mechanism described in the previous section, it can spread a pheromone identifying such objects into the available memory of near *location-tags*.
4. This enables to spread pheromone trails associated with the objects across the *location-tags* of the environment.
5. When looking for an object, a user can instruct the agent to read in-range *location-tags* searching the object’s pheromone ID in the tags memories. This requires that

the agent can locally access a table associating each object to a specific pheromone ID.

6. When the pheromone trail of the searched object is found, the user can follow it uphill to reach the object current location.

Once the object has been reached, if it starts moving with the user (i.e. the user has grabbed it), the application automatically starts spreading again the pheromone associated with the object, to keep consistency with the new object location.

This application naturally suits a multi-user scenario where a user (or a robot), looking for an object moved by another user, can suddenly cross the pheromone trail the object left. Also, in the presence of multiple users/robots, one could effectively exploit parasitic diffusion of pheromones to increase the probability to cross the pheromone trail being searched.

5.1 Spreading Object Pheromones

The spreading of pheromones in this application requires the agent to understand which objects are currently being carried (i.e. moved around) by its user. To perform this task unobtrusively, it accesses the RFID reader to detect in-range RFID tags once a second.

Let us call $O(t)$ the set of *object-tags* being sensed at time t, and recall that $L(t)$ is the set of *location-tags* being sensed at time t. If the agent senses an *object-tag* O such that $O \in O(t)$, $O \in O(t-1)$, but $L(t) \neq L(t-1)$, then the agent can infer that the user picked-up the object O and the object is moving around. In this situation, the agent has to spread O pheromone in the new location. To this end, the agent writes O in the available memory space of all the $L(t)$ *location-tags* that do not already contain O. This operation is performed, for every object O, upon every subsequent movement. Similarly, if the agent senses that an *object-tag* $O \in O(t-1)$, but $O \notin O(t)$, then the agent infers that the user left the object O. When this situation is detected the agent stops spreading the O pheromone. These operations create pheromone trails of the object being moved around.

It is worth noticing that the presented algorithm works best for rather short range readers, where if the above conditions are met, the object has been truly picked up. In the case of long range readers, it can happen that the areas covered by two subsequent readings overlap. If an object lies in the intersection of two of these areas, the algorithm infers that the object has been picked up, while it was not touched. This creates a spurious object pheromone near the object itself. However, for the sake of the object tracking application, this is not a big problem, in that the pheromone is propagated only in the close proximity to where the object is actually located.

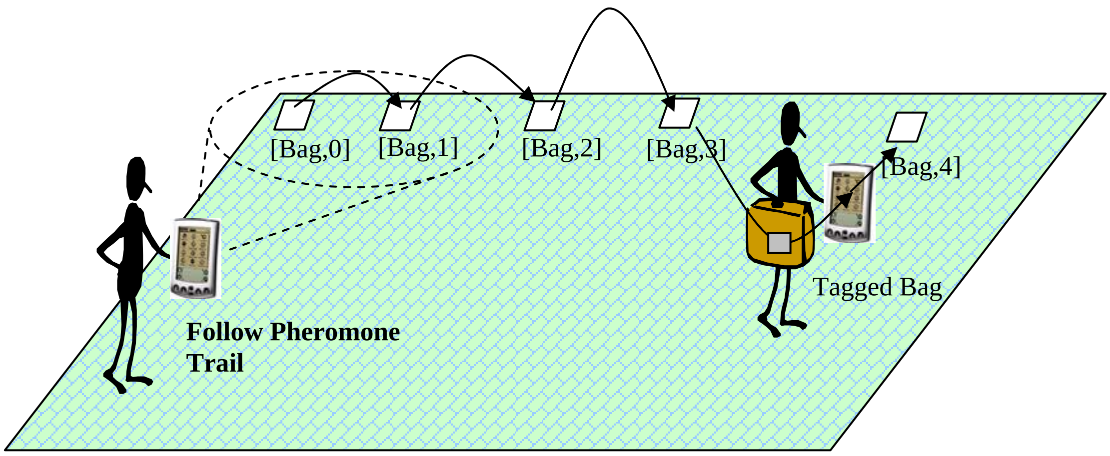


Figure 5. Object-tracking. As the user with the bag moves around, the agent on the PDA recognizes the user is carrying a tagged bag, and then spread the "Bag"-pheromone (and the associated hop counter) in the location-tags nearby. Meanwhile or later on, another user can look for the bag by following the "Bag" pheromone trail.

5.2 Tracking Objects

Once requested to track an object *O* the agent will start reading, once per second, nearby *location-tags* looking for an *O*-pheromone within the sensed *location-tags L(t)*. If such a pheromone is found, this implies that the user crossed a suitable pheromone trail. Then, this trail has to be followed uphill (in the direction of increasing *C(O)*) to find the object. To this end, two alternatives arise: either *L(t)* contains only one *location-tag* (as it may happen with very short-range RFID readers) or *L(t)* contains at least two *location-tags* having *O*-pheromones with different *C(O)* (as it may happens with medium- and long-range RFID readers).

In the former (unlucky) case, the application notifies the user about the fact he has crossed a pheromone trail, but nothing else. In such situation, the user has to move in the neighborhood, trying to find higher *C(O)* indicating the right direction to be followed (this is like dowsing -i.e. finding underground water with a forked stick – but it works!). We refer to this as *local-search.*

In the latter (lucky) case, the agent notifies the user about the fact he has crossed a pheromone trail and it suggests to move towards those *location-tags* having the higher *C(O)*. In the following, we will refer to this as *grad-search*, since it is like following a gradient uphill. With this regard, it is important to emphasize that *grad-search* is likely to be available only with RFID readers with a range long enough to include in *L(t)* at least

two tags storing the pheromone trail. Moreover, since we do not require the presence of localization devices, the agent suggests the user to get closer to the location having higher *C(O)*, by naming the location – e.g., walk to the "front door" – and the user has to know how to get there without further help (or, for robots, it has to internally code a map of the locations in the building).

In either cases, following the agent advices, the user gets closer and closer to the object by following its pheromone trail, until reaching it.

6. EXPERIMENTS

To assess the validity of our approach and the effectiveness of the object tracking application, we performed a number of experiments, both adopting a real implementation and a simulation framework. Basically, our approach consisted in testing the feasibility and usability of the system on the real implementation, and then in developing a simulation framework matching the real data in large-scale scenarios.

6.1 Real Implementation Set-Up

The real implementation consisted in tagging places and objects within our department (Figure 6a). Overall, we tagged 100 locations within the building (doors, hallways, corridors, desks, etc.) and several objects (books, laptops, cd-cases, etc.) within. Locations have been tagged with ISO15693 RFID tags, each with a storage capacity of 512 bits (each tag contains 30 writable byte slots, and can store 6 pheromones overall). Objects have been tagged with ISO14443B RFID tags, each with a storage capacity of 176 bits (each tag contains only the object ID and two key-value pairs).

For users, we exploited HP IPAQ 36xx PDAs, each running Familiar Linux 0.72, J2ME (CVM – Personal Profile), and provided with a WLAN card and an Inside M21xH RFID reader (Figure 6b), which is a medium-range one (making it possible to read tags at a distance up to 20 cm).

To test the effectiveness of our approach, we made several experiments organized as follows. One user carries on an object, hides it within the department and deploys a corresponding pheromone trail. Later on, two users try to find the object. One of them without any support or suggestion, another trying to identify and follow the pheromone trail associated to the object. What we found is that in the majority of the cases, the user following the pheromone trail was able to find the object faster. However, we noticed that the process of following the pheromone was notably slowed down because of the rather

limited range of the adopted RFID reader, implying the user to adopt a *local-search* (i.e., wandering around the trail to identify the uphill direction).

Figure 6. (a) Some tagged objects. (b) The test-bed PDA hardware. (c) The Lego Mindstorms robots with a PDA and an RFID reader mounted aboard.

In addition, to test the effectiveness of our approach for robots other than for humans, another set of experiments have been realized by installing one PDA connected to a RFID reader onboard of Lego Mindstorms robots (*www.legomindstorms.com*). The IPAQ runs an agent controlling both the RFID reader and the robot microprocessor (Figure 6c). 100 (10x10) RFID tags have been attached to the pavement grid, and robots were to find a specific tag ID in the grid. We then compared the behavior of a robot exploring the grid blindly (without pheromone support) and that of a robot following pheromone trails previously spread. Also in this case, the limited range of RFID readers makes *local-search* the only possible solution. What we found here (in contrast with the experiment performed with human users), is that the robot following the pheromone trails, was not able to significantly outperform the robot exploring the grid blindly.

On the basis of this contrasting experiment, and to better unfold the effectiveness of our systems and the effects of the various parameters involved, we also set up a set of simulation experiments.

6.2 Simulation Set-Up

To test more extensively and on the large scale, we realized a JAVA-based simulation of the above scenario. The simulation is based on a random graph of places (each associated to a *location-tag*), and on a number of objects (each associated to an *object-tag*) randomly deployed in the locations-graph. Each tag has been simply simulated by an array of integer values.

A number of agents are simulated wandering randomly across the locations-graph, collecting objects, releasing objects, and spreading pheromones accordingly. At the same time, other simulated agents are looking for objects in the environment eventually exploiting pheromone trails previously laid down by other agents.

The simulator allows performing a number of experiments changing a number of parameters such as the graph size, the number of objects, the number of agents involved, the storage capacity of the tags, etc. For the sake of comparison, we tested both the *local-search* algorithm in which the agents perceive the pheromones in their current node, but cannot see the direction in which the pheromones increase, and the *grad-search* algorithm, in which the agents perceive pheromones together with the directions in which they increase. Also, these have been compared with a *blind-search* algorithm, in which agents explore the environment systematically, fully disregarding pheromones.

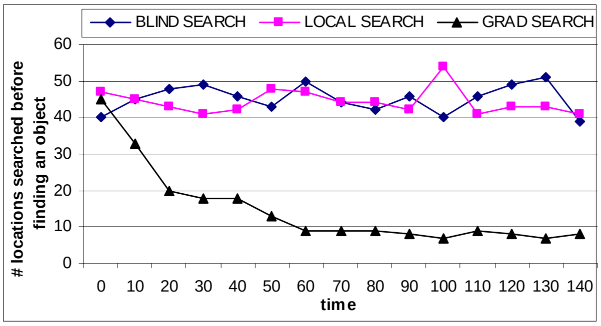
6.3 Results of the Simulation Experiments

A first group of experiments (Figure 7) aims at verifying the general effectiveness of our approach and of the object-tracking application.

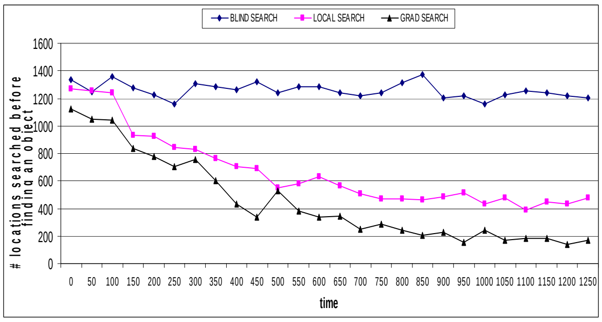
We report results from two different simulation scenarios: the first consisting in 100 tagged places with 100 objects (Figure 7-a); the second consisting in 2500 tagged places with 500 objects (Figure 7-b). A number of 10 agents have been simulated to populate these environments wandering around moving objects and spreading pheromones (only proactively – the effects of parasitic diffusion will be discussed later on) and, at the same time, looking for specific objects. In the experiments, we report the number of places visited (i.e., the number of *location-tags* perceived) before finding specific objects, for different search methods, plotted over a virtual time. The reported results are averaged of about 300 simulations.

Starting from a scenario free of pheromones (time zero in Figure 7-a), the more time passes the more pheromone trails get deployed by agents. *Blind-search* does not take advantage of pheromone trails: objects are found after visiting on the average half of the places. *Grad-search* takes a great advantage of pheromones: after an initial period, and when several pheromone trails have been deployed, *grad-search* starts becoming very effective: less than 10% of the places need to be visited before finding the object. *Local-search*, instead, appears not to take any relevant advantage of pheromones. This is due to the cost of orienting in the environment to find the proper direction, at least in the small-scale scenario of Figure 7-a. These results are perfectly in line with the experiments performed on the Lego robot that, by adopting *local-search* in a grid of 100 tags, were not able to significantly take advantage of pheromones. The fact that, instead, humans can take advantage of pheromones even with *local-search*, derives from the fact the actual topology of a building (e.g., our department) is generally more constrained than a random graph or a grid. Thus, humans does not require to repeat the process of finding the uphill directions at each and every step.

In any case, such a situation changes when getting to larger-scale scenarios (as in Figure 7-b). There, both *local-search* and *grad-search* appears reasonably effective. The performance improvements of *local-search* in this case are due to the fact that the cost of "orienting" in a local neighborhood becomes negligible when the environment is large. Thus, although the *grad-search* algorithm is always preferable, in large-scale scenarios our approach is effective even when using short-range reader enabling to enforce the *local-search* algorithm only.



a)



b)

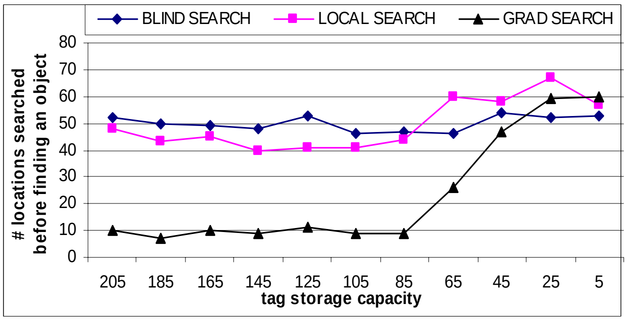
Figure 7. Number of places visited before finding a specific object plotted over time. (a) 100 tagged places. (b) 2500 tagged places.

A second group of experiments aims at exploring the effects of RFID tag storage saturation upon pheromone spread (Figure 8). This of course represents a big problem, in fact, it can happen that pheromone trails can be interrupted, because there is not available space left on neighbor *location-tags*, while the object to be tracked moves away. This creates a broken pheromone trail leading to a place that is not the actual location of the object.

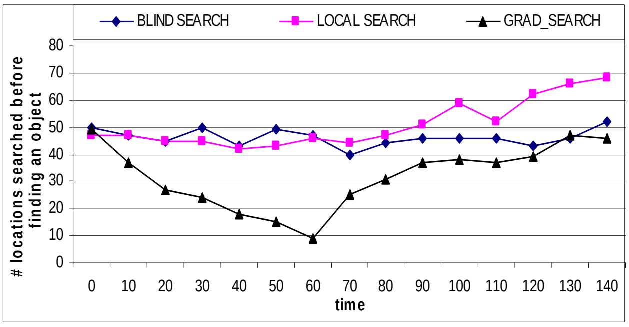
In Figure 8-a, we report an experiment conducted in the 100-tagged-places-environment described before. We plot the number of places visited before finding specific objects for different search methods, over a shrinking tag storage capacity. In these experiments, agents move carrying objects (and thus spreading pheromones) for 150 time steps, then they start looking for objects without picking up any of them (and thus without spreading pheromones anymore). This set up is intended to freeze the pheromone deployment after a certain time (150 steps) and better evaluate the role of tag storage space.

Let us focus on the *grad-search* method that is the most interesting in this context. It can be noticed that, when the tag storage capacity is high, we have good performance. However, when the capacity fall below 85 pheromones (that is – when the tag has a capacity of less than 85 * 3 bytes = 255 bytes), performance starts decaying really fast

and when the capacity is lower than 25 (75 bytes), *grad-search* works equal to *blind-search*. It is rather easy to explain this phenomenon: when the tag capacity is low, there are a lot of broken pheromone paths degrading the performances. An agent, reaching the end of a broken pheromone trail, has no choice but starting the search from the beginning.



a)



b)

Figure 8. (a) Number of places visited before finding a specific object plotted over a shrinking tag storage. (b) Number of visited places before finding a specific object plotted over time, when tags tend to saturate.

Figure 8-b shows the same problem from the time perspective, with the tag capacity fixed to 50 pheromones (150 bytes). The experiment shows the number of places visited before finding specific objects, for different search methods, over time. Let us focus again on the *grad-search* behavior. It is easy to see that, when time is close to zero, *grad-search* works equal to *blind-search*, since no pheromone trails have been laid down. After some time, *grad-search* works considerably better than *blind-search*, since pheromone trails drive agents. However, as time passes, tags capacity tends to saturate: the objects are moved, but no pheromone trails can be deployed. This situation rapidly trashes performance leading back to *blind-search* performance. For instance, in our real

implementation (tags with a 512 bits capacity), the above problem leads to a large number of broken trails as soon as more than 20-30 objects are being tracked.

Finally, in the experiments depicted in Figure 9, we tried to assess whether the pheromone evaporation mechanism can help in such situation. Figure 9 plots the number of places visited before finding specific objects over a shrinking tag storage capacity (the same of Figure 9-a). This time, however, only *grad-searches* are depicted and each plot is associated to a different threshold *T* of pheromone evaporation. Unfortunately, it is rather easy to see that the pheromone evaporation is rather ineffective.

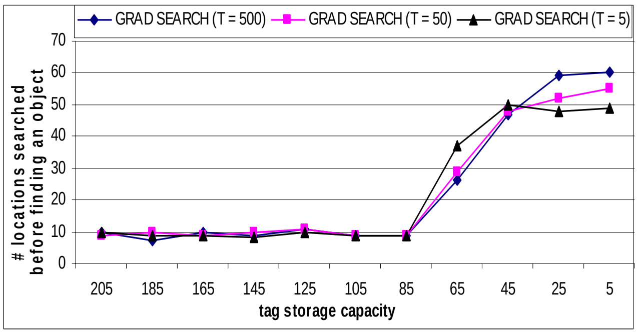


Figure 9. Number of visited places before finding a specific object plotted over a shrinking tag storage space, for various evaporation threshold.

6.4 Effects of Parasitic Diffusion

When pheromones are allowed to diffuse not only in a proactive way (as in the experiments above) but also parasitically, two effects arise:

- The number of visited places before a pheromone trail is encountered diminishes, which contributes scaling down the time needed to find objects, both for *local-search* and for *grad-search*. Again, this effect is more relevant in large-scale scenarios.
- The effect of memory saturation worsen, due to the increased number of pheromone trails that have to be stored around in the environment.

Clearly, the above two effects are increasingly evident the more agents exists in the environment that parasitically diffuse pheromones.

6.5 Discussion

The above experiments shows that our approach is effective but it also exhibits problems.

From the positive side, our approach is effective in deploying pheromone trails in the environment. This can be used to enable stigmergic coordination and to achieve context-awareness. As the experiments in the object-tracking application show, such pheromone trails can be effectively exploited by agents whether they exploit *local-search* (requiring cheap and small short-range RFID readers) or *grad-search* (requiring larger RFID readers), though the latter remains preferable.

From the negative side, the main problem of our approach relates to the limited storage capacity of the RFID tags. Basically, if the number of objects to be tracked (or, more in general, if the number of pheromone trails to be deployed in an environment) is greater than the available slots on the RFID tag, in the long run the problem is unavoidable. Sooner or later, a new object will cross to an already full tag, breaking the pheromone trail. The pheromone evaporation mechanism that we implemented did not help this situation.

We still do not have a solution for this problem. Our research with regard to this topic is leading in two main directions: *(i)* we are currently researching more advanced pheromone evaporation mechanisms. *(ii)* We are considering the idea of spreading pheromone trails not only in *location-tags* but also on *object-tags*. The advantage would be that the more objects are in the system, the more storage space is available for pheromones, letting the system to scale naturally. The problem is how to manage the fact that *object-tags* containing pheromones can be moved around, breaking the pheromone trail structure. As a partial relief from this problem, it is worth reporting that (as already anticipated) recent RFID tags have a storage capacity in the order of several of KB, and the trends indicate that such capacity will further increase in the near future. This will make it possible to track hundreds of objects (or, more in general, to spread a large amount of pheromone trails in an environment) without changing our approach.

7. OTHER APPLICATION SCENARIOS

Pheromone interaction and stigmergy have attracted more and more researches due to their power in supporting context-awareness and adaptive coordination in a variety of scenarios. Thus, it is not surprising that even our proposal for RFID pheromone deployment could find a number of additional applications, beside the presented object-tracking application.

In general, the value and novelty of RFID-based infrastructures can be best perceived in the context of challenged scenarios where usual infrastructures based on network connectivity may not be available [MamQZ06, Swe06]. In these situations, RFID tags

provide an uniquely cheap, easily deployable and likely-to-be-already-there infrastructure.

More in concrete , one could think at exploiting RFID pheromones to enable a group of users and robots to coordinate on-the-fly and without any advanced planning their movement in an unknown and challenged environment. Consider for example an emergency rescue team (whether human, robotic, or mixed) arriving in a disaster area where no computing/network infrastructure is available other than the nearly unbreakable RFID tags. One the one hand, if the team members exploit these tags to spread pheromones around as they walk, and are instructed to stay away from existing pheromone trails, then one can have reasonable guarantees that the whole environment is explored in a comprehensive and effective way by the group [SveK04]. On the other hand, whenever a member of the rescue team discover something important that requires to be found by other members of the team (e.g., a robot finding a injured person that require medical assistance), it can start spreading a pheromone leading to that something, so that other members (e.g., first-aid doctors passing by) can notice it.

In this context, it is important to remark that our approach clearly requires the presence of RFID tags before pheromones can be spread. Although RFID tags are likely to be soon densely present in everywhere (embedded in tiles, bricks, furniture, etc.), one cannot rely on this in sensible situations like the actions of a rescue team. In these cases, however, it is possible to conceive solutions where users or robots physically deploy RFID tags on-the-fly while exploring the environment, to be used for subsequent coordination. Also, future development in plastic (and printable) RFID technology [Col04] let us envision the possibility of enriching an RFID reader with a simple RFID printer to dynamically print RFID tags in pavements, walls, or any type of surface, whenever needed.

In line with these ideas is the concept of pervasive workflow management. Standard workflow management systems are rooted on a software engine keeping track of the status of the workflow being carried on. Workers notify to this engine the tasks being completed and the engine in turn notifies the subsequent tasks that have to be carried on. RFID tags and pheromone-based interaction could remove the need for a centralized engine in pervasive computing environments and lead to more situated and adaptive scenario. For instance, in a disaster scenario, first-aid rescuers could tag injured people and write in the tags information specifying the first diagnosis. Subsequently, physicians could take advantage of such information to provide a responsive medication services.

More in general, RFID tags can be used to help users (as well as robots) in getting aware of what's in the environment more than their natural and artificial senses can do, i.e., by reading additional information that can be provided by tags. To some extent, RFID tags may enable a sixth "digital" sense with which to gather digital information from locations and objects. While this can be a simply add-on for people with normal abilities, the additional sensing capabilities provided by RFID tags may be dramatically important for, e.g., helping visually impaired in getting aware of what's around [Kul04]. Specifically, the use of pheromones trails can support a guided navigation of visually impaired towards specific locations/objects in the environment (we specifically envision a visually impaired person having mounted a short-range RFID reader on its white stick and sensing pheromone trails stored on RFID tags attached to the pavement).

Other than for navigation purposes, the activities of accessing (or simply getting in range) with some RFID tags can be used to achieve awareness of the activities occurring in the environment. One of the most interesting works in this direction has been presented in [Phi04]: a software application is able to infer the users' daily activities on the basis of the objects he touches (e.g. if the user touches a teapot and a cup, the application can infer that he is preparing tea). All these facets of context-awareness – which mostly exploit information assumed to be already stored in tags, can be enriched by the ideas presented in this paper, suggesting to: *(i)* exploit RFID tags in the environment as a sort of distributed shared memory for some history of locally occurred events; *(ii)* exploit pheromones to keep a traceable distributed track of past environmental activities. For instance, in the application for inferring daily activities, one could think that – once the application recognizes that some tea is being prepared, the teapot start spreading a pheromone trail leading to the fridge and indicating that some tea has already been prepared and is there to cool down.

8. RELATED WORK

In the last few years, a lot of distributed applications inspired by pheromone interactions have been proposed. However, only a few of them, though, define actual solutions to deploy pheromone-coordinated systems in pervasive and mobile computing scenarios.

In the absence of a physical infrastructure on which to deploy pheromones, a possible solution is to provide a virtual representation of the environment and of the pheromone trails. For instance, a pheromone-based approach to coordinate swarms of unmanned airspace vehicles (UAVs) has been implemented and tested on simple prototype UAVs,

in which pheromones are spread in virtual environment accessed as a sort of distributed shared memory by all UAVs [ParBS05]. As explained in the paper, the novelty of our proposal is that it works also in situation where network connectivity is absent. In any case, nothing prevents to combine our approach with those using a virtual representation of the pheromone landscape. In particular, the two approaches could be combined so as to use RFID when the network is unavailable, and suitable pheromone server when the network is present. Such kind of mixed approaches would go in the direction of autonomic computing, where the system self-adapts to changing environments to provide a suitable service in any case.

Another proposal in the literature consists of spreading pheromones as an overlay distributed structure over the ad-hoc network of the agents to be coordinated. For instance, some proposals for swarm robotics [LurS04] suggest spreading and diffusing pheromones (e.g., pheromone trails) over the ad-hoc network defined by the robots themselves. Such solution presents problems related to the cost of individual robots, and on the number of robots required to provide a good coverage of the environment and a dense enough network. Also, should the ad-hoc network of robots get partitioned, pheromone trails would be broken. For this reason, the solution appears most suitable for coordinating activities in modular robots and self-assembly, where the density of the network is ensured by the direct contact between components [SheS02].

In any case, also with regard to this proposal a suitable combination of RFID and overlay network could provide better systems. For example the RFID tags could be used as a cache to let the system survive connectivity problems.

If one could assume the presence of a pervasive network infrastructure, i.e., a wireless sensor network deployed in the environment, then one could effectively use the nodes of such network to store and access pheromones, as described by Li and colleagues [LirRR03]. This approach is somewhat similar to our RFID implementation: agents connect to nearby sensors and store pheromones in there. In the long-term this would be the most powerful solution: sensors, being active, are more reliable and could implement pheromone evaporation autonomously. However, such solution is presently very costly. Also, wireless sensor networks exhibit battery-exhaustions problems (and, thus, limited life) which the RFID solution prevents.

Clearly, the solution that would most closely mimic the actual behavior of social insects would be that of physically releasing markers in an environment. For instance, Svennebring and Koenig [SveK04] have implemented and tested – for the sake of terrain

exploration – robots equipped with a pen to leave special metal ink trails in the pavement and a sensor to sense such trails. In this way, a group of robots can enforce a simple form of pheromone-based coordination (e.g., if an ink trail is sensed by a robot, it means that another robot has already covered that part of the terrain). Apart from the fact that spreading ink around is not a nice and easily acceptable solution for pervasive computing scenarios, the RFID tag solution is much more flexible, in that it enables using more semantic information for a wider range of applications.

An interesting approach that exploits RFID tags to enforce a sort of marker-based coordination in the activities of robots devoted to collective construct composite artifacts is described in [Wer06]. On the one hand, robots can acquire awareness about the current positioning and nature of the building blocks by reading location information embedded in RFID tags attached to them. On the other hand, robots that move blocks can write updated location information on them, for other robots in the team to continue the cooperative construction work in a coordinated way. Although focused on a rather different application scenario, such RFID-based form of coordination definitely confirms the suitability of RFID technology to act as a general substrate to enforce indirect and pheromone-based forms of interactions in the physical world.

9. CONCLUSIONS AND FUTURE WORK

RFID technology, whose effectiveness in improving our interactions with the physical world have already been proved in a variety of pervasive computing projects, also represents a flexible and low-cost way to take advantage of the power and simplicity of pheromone-based interaction models. In particular, the approach presented in this paper exploits RFID tags as a sort of distributed environmental infrastructure that mobile autonomous agents – whether humans or robots – can exploit both to spread and sense digital pheromones. In this way, agents can adaptively acquire context-awareness and coordinate with each other in a very simple way, features which can be fruitfully exploited in a variety of application scenarios.

While a preliminary prototype implementation already shows the feasibility of our approach, a number of research directions are still open to improve its practical applicability. First, more experiments are required to better verify the scalability of the proposed approach to very large-scale scenarios and in the presence of a large number of users. In particular, based on the limitations identified in this paper, effective solutions must be found to the problems related to broken pheromone trails and pheromone evaporation. Second, we need to explore the possibility of extending our strictly local and

environment-centered approach (the only exploited information is that available in RFIDs) with the possibility of accessing additional information made available by some sort of “pheromone” servers. The coupling of local RFID pheromone information with some more global information, without undermining the advantages of our approach, can enable to reach higher degrees of context-awareness whenever a network connection is available. Third, we are perfectly aware that the use of RFID to spread and access distributed information in an environment raises serious privacy and security concerns [Sta05], currently suggesting limiting the use of our system within controlled indoor environment. However, since a number of diverse approaches are being analyzed and proposed to tackle these issues [RieCT06], we are confident suitable solutions will be soon available for integration in our approach.

Finally, some theoretic analysis on the general properties of pheromone propagation in RFID environments (e.g., computing the likelihood of broken trails under different hypothesis) could provide important general results on the behavior and performance of our approach. In particular, comparisons with game-theory algorithms [SavM99] could provide useful insights.

REFERENCES

[BabMM02] O. Babaoglu, H. Meling, A. Montresor, “A Framework for the Development of Agent-Based Peer-to-Peer Systems”, Proceedings of the IEEE International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press, May 2002.

[BonDT99] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence*, Oxford University Press (Oxford, UK), 1999.

[Bor04] C. Bornhovd, T. Lin, S. Haller, J. Schaper, “Integrating Automatic Data Acquisition with Business Processes: Experiences with SAP’s Auto-ID Infrastructure”, Proceedings of the 30th International Conference on Very Large Databases, Toronto (CA), IEEE CS Press, pp. 1182-1188, 2004.

[Col04] G. Collins, “Next Stretch for Plastic Electronics”, *Scientific American*, 291:74-81, August 2004.

[FloL05] C. Floerkemeier, M. Lampe, “RFID middleware design - addressing application requirements and RFID”, Technical Report ETH Zurich, Pervasive Computing Group, Submitted for Publication, 2005.

[Hah04] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, M. Philipose, "Mapping and Localization with RFID Technology", *International Conference on Robotics and Automation*, Barcelona (ES), IEEE Press, April 2004.

[HsiF05] S. Hsi, H. Fait, "RFID Enhances Visitors' Experience at the Exploratorium", Communications of the ACM, 48(9):60-65, Sept. 2005.

[KleP06]A. Kleiner, J. Prediger and B. Nebel, "RFID Technology-based Exploration and SLAM for Search And Rescue", *International Conference on Intelligent Robots and Systems*, Beijing, China, IEEE Press, 2006.

[Kul04] V. Kulyukin, C. Gharpure, J. Nicholson, S. Pavithran, "RFID in Robot-Assisted Indoor Navigation for Visually Impaired", *International Conference on Intelligent Robots and Systems*, Sendai (J), IEEE Press, 2004.

[LiRR03] Q. Li, M. De Rosa, D. Rus, "Distributed algorithms for guiding navigation across a sensor network", *International Conference on Mobile Computing and Networking*, San Diego (CA), ACM Press, October 2003.

[LurS04] J. McLurkin, J. Smith, "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots", *International Symposium on Distributed Autonomous Robotic Systems*, Toulouse (F), IEEE CS Press, June 2004.

[MamZ05] M. Mamei, F. Zambonelli, "Spreading Pheromones in Everyday Environments Through RFID Technology", *Symposium on Swarm Intelligence*, Pasadena (CA), IEEE Press, pp. 281-288, June 2005.

[MenT03] R. Menezes, R. Tolksdorf, "A New Approach to Scalable Linda-systems Based on Swarms", Proceedings of the 18th ACM Symposium on Applied Computing, Orlando, FL (USA), ACM Press, March 2003.

[Par97] V. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems", *Annals of Operations Research*, 75:69-101, 1997.

[ParBS05] V. Parunak, S. Brueckner, J. Sauter, "Digital Pheromones for Coordination of Unmanned Vehicles", in *Environments for Multi-agent Systems*, LNCS No. 3374, Springer Verlag, 2005.

[Phi04] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hahnel, "Inferring Activities from Interactions with Objects", *IEEE Pervasive Computing*, 3(4):50-57, 2004.

[MamQZ06] M. Mamei, R. Quaglieri, F. Zambonelli, "Making Tuple Spaces Physical with RFID Tags", *Symposium on Applied Computing*, Dijon (F), ACM Press, April 2006.

[RieCT06] M. R. Rieback, B. Crispo, A. S. Tanenbaum, "The Evolution of RFID Security", *IEEE Pervasive Computing*, 5(1):62-69, Jan.-March 2006.

[Sat05] I. Satoh, "A Location Model for Pervasive Computing Environments", *International Conference on Pervasive Computing and Communications*, Kauai Island (HW), IEEE CS Press, March 2005.

[SavM99] R. Savit, R. Manuca, and R. Riolo, "Adaptive Competition, Market Efficiency, and Phase Transitions". *Physical Review Letters*, 82(10):2203-2206, 1999.

[SheS02] W. Shen, B. Salemi, P. Will, "Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots", *IEEE Transactions on Robotics and Automation,* 18(5):1-12, Oct. 2002.

[Sta05] F. Stajano, "RFID Is X-Ray Vision", *Communications of the ACM*, 48(9):31-33, Sept. 2005.

[SveK04] J. Svennebring, S. Koenig, "Building Terrain Covering Ant Robots: a Feasibility Study", *Autonomous Robots*, 16 (3):313-332, May 2004.

[Swe06] C. Swedberg, "First Responders Can Tag Victims For Tracking", RFID Journal, www.rfidjournal.com/article/articleview/2708/1/1, 2006.

[Wan04] R. Want, "Enabling Ubiquitous Sensing with RFID", *IEEE Computer*, 37(4):84-86, April, 2004.

[Wan06] R. Want, "An Introduction to RFID Technology", *IEEE Pervasive Computing*, 5(1):25-33, Jan.-March 2006.

[Wer06] J. Werfel, Y. Bar-Yam, D. Rus, R. Nagpal, "Distributed Construction by Mobile Robots with Enhanced Building Blocks", *International Conference on Robotics and Automation*, Orlando (FL), IEEE Press, May 2006.